HPSS **T**ape **Ar**chiver (**htar**)

Kim Cupps

Lawrence Livermore National Laboratory

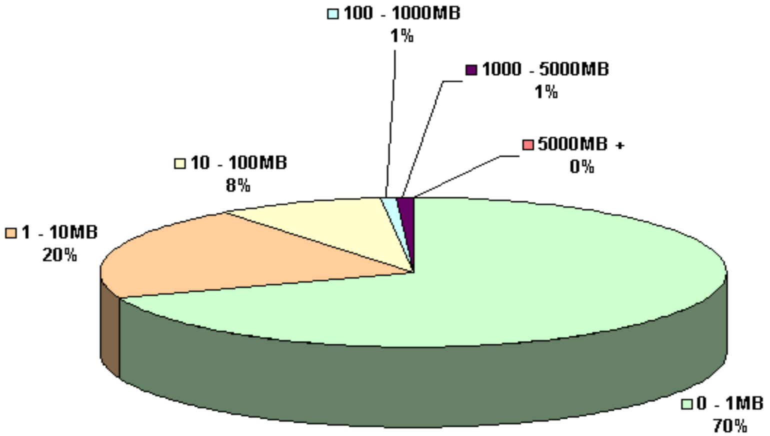
June 5, 2001

Outline

- **Statement of the problem**
- **Problem mitigation: htar**
- **Overview of htar**
  - ? Functionality highlights
  - ? Usage examples
  - ? Implementation details
  - ? Buffering scheme block diagram
  - ? Performance testing results
- **Future work**

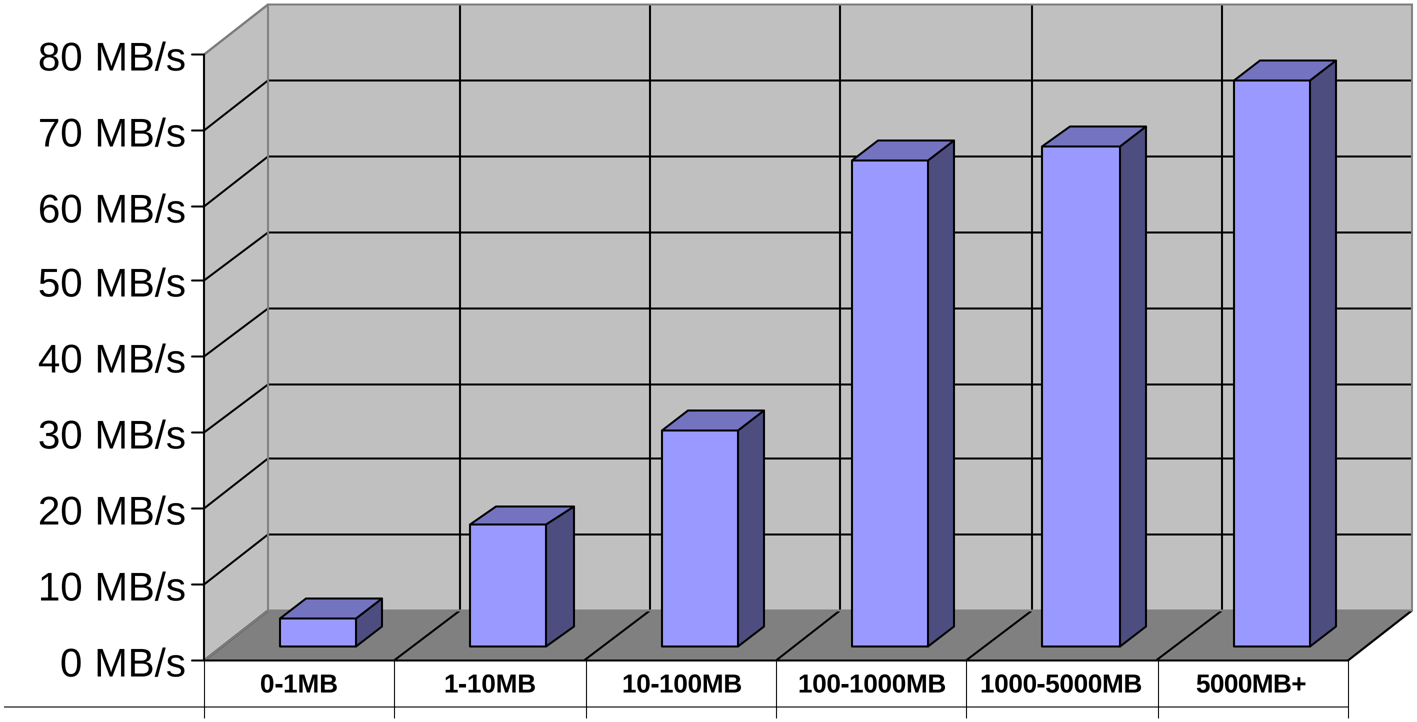
Problem statement

- **The large majority of files transferred to our systems are less than 1 MB**
  - ? Users transfer hundreds or thousands of small files
  - ? We lose HPSS performance opportunities
  - ? Lots of metadata to manage



Problem statement

- **pftp write performance improves dramatically with file size**



Problem mitigation: htar

? htar is a multi-threaded, on-the-fly file bundling and data transfer tool created by Mike Gleicher for use with HPSS

? htar uses the POSIX 1003.1 tar file format for the bundled file

? htar was built based on the following set of requirements

» The data bundling utility will be optimized for the GPFS file system running on LLNL's ASCI white platform

» There will be a mechanism to list the contents of a bundle (files) after it has been created

» It is preferred that users can retrieve individual files from a bundle stored in HPSS without retrieving the entire contents of the bundle

» HPSS will not be aware of the bundle concept and will treat a bundle as a file

» A standard file format is required

» A file bundling prototype (proof of concept) shall be delivered to LLNL's testbed platform no later than November 15th, 2000

Htar overview

- ? Some htar definitions
  - » Archive file – the large file that htar creates out of the small files
  - » Member file – one of the files contained in the htar archive file
  - » Index file – a directory of the member files contained in the archive file
  - » Consistency file – the file at the end of an archive file used to verify the consistency of the archive file and the index file
- ? Htar functionality highlights
  - » htar creates an index file that is used to specify the list of files in the archive
  - » htar can be used to retrieve a single file from an archive file
  - » htar can create an index file for an existing tar file
  - » The index file is used for all htar operations except “build index”

A few htar usage examples

? Bundles the local files file1, file2 and file3 into an archive file called kimsfiles.tar and stores them in HPSS home directory

» htar –cf kimsfiles.tar file1 file2 file3

? Displays the names of the files in the kimsfiles.tar in the HPSS home directory

» htar –tf kimsfiles.tar

? Extracts file2 from kimsfiles.tar, retrieves file2 from storage, and writes file2 to the current local directory

» htar –xf kimsfiles.tar file2

? Builds an index file for the tar file called oldfile.tar in the HPSS home directory

» htar –Xf oldfile.tar

? Bundles the local files file1, file2 and file3 into an archive file called kimsfiles.tar and stores them in the local directory

» htar –cEf kimsfiles.tar file1 file2 file3

Implementation details

? Htar uses multiple threads to read local disk, construct the archive file, and transfer the file to HPSS

? A coordinated buffering scheme is used to manage phases of data collation and data transfer

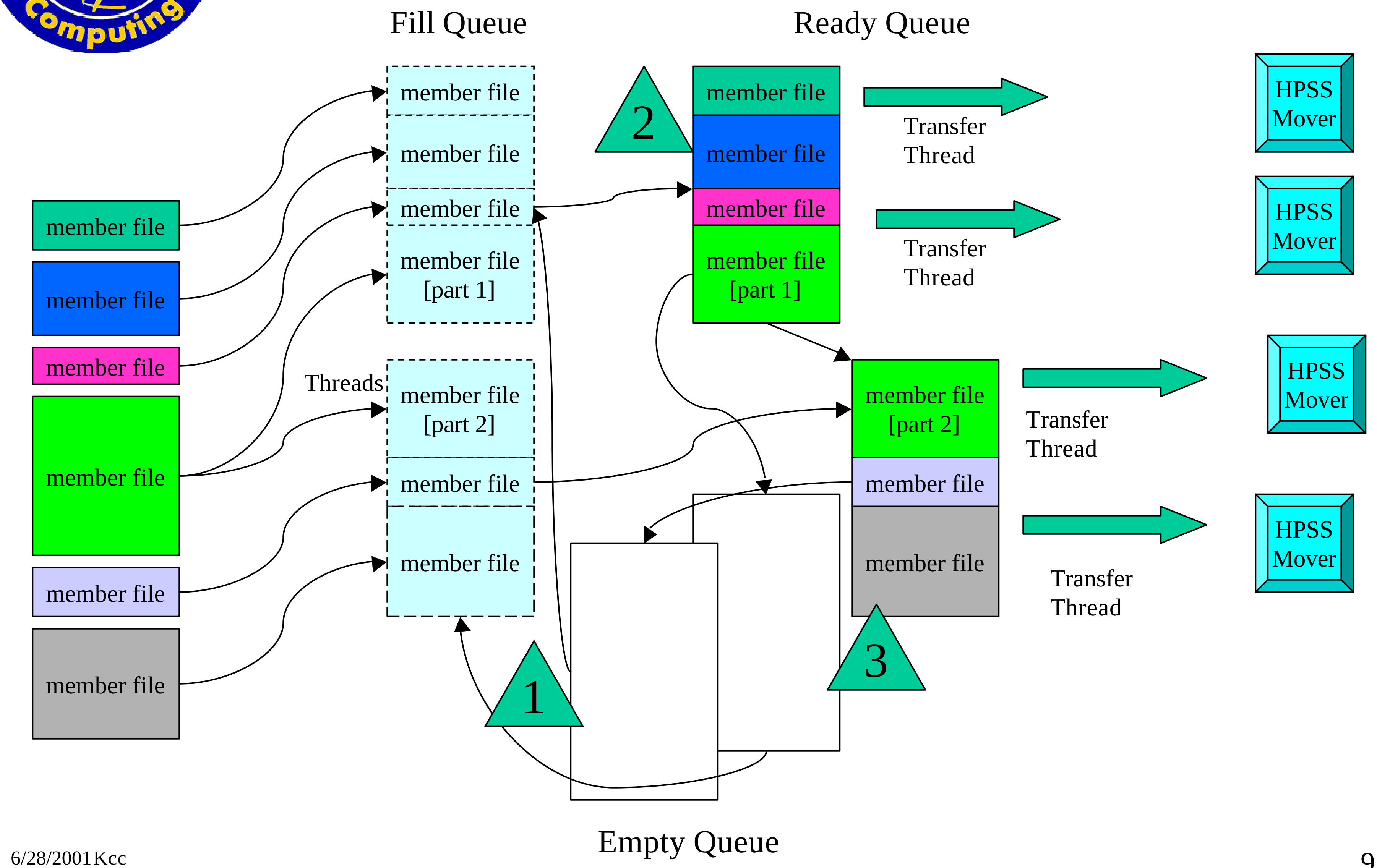
? Each index file is 512 bytes per member file plus a fixed header – a 10,000 file archive has a 5MB index file

? Htar runs on AIX (DCE and non-DCE), Tru64 UNIX, IRIX, and Solaris

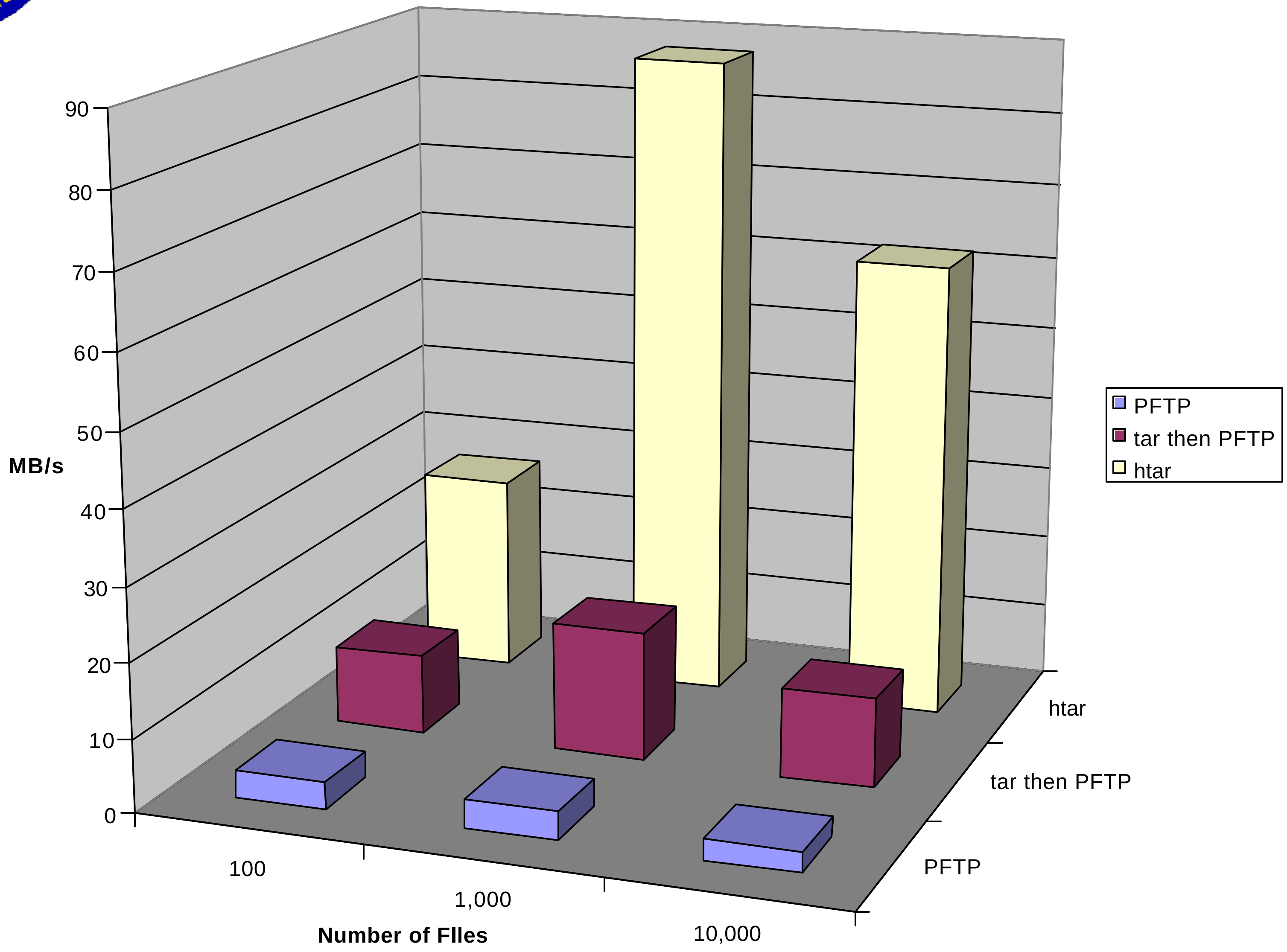
? On HPSS platforms htar is built using the HPSS DCE-based client API library

? To run htar on non-HPSS platforms, it must be built using the HSI non-DCE client library

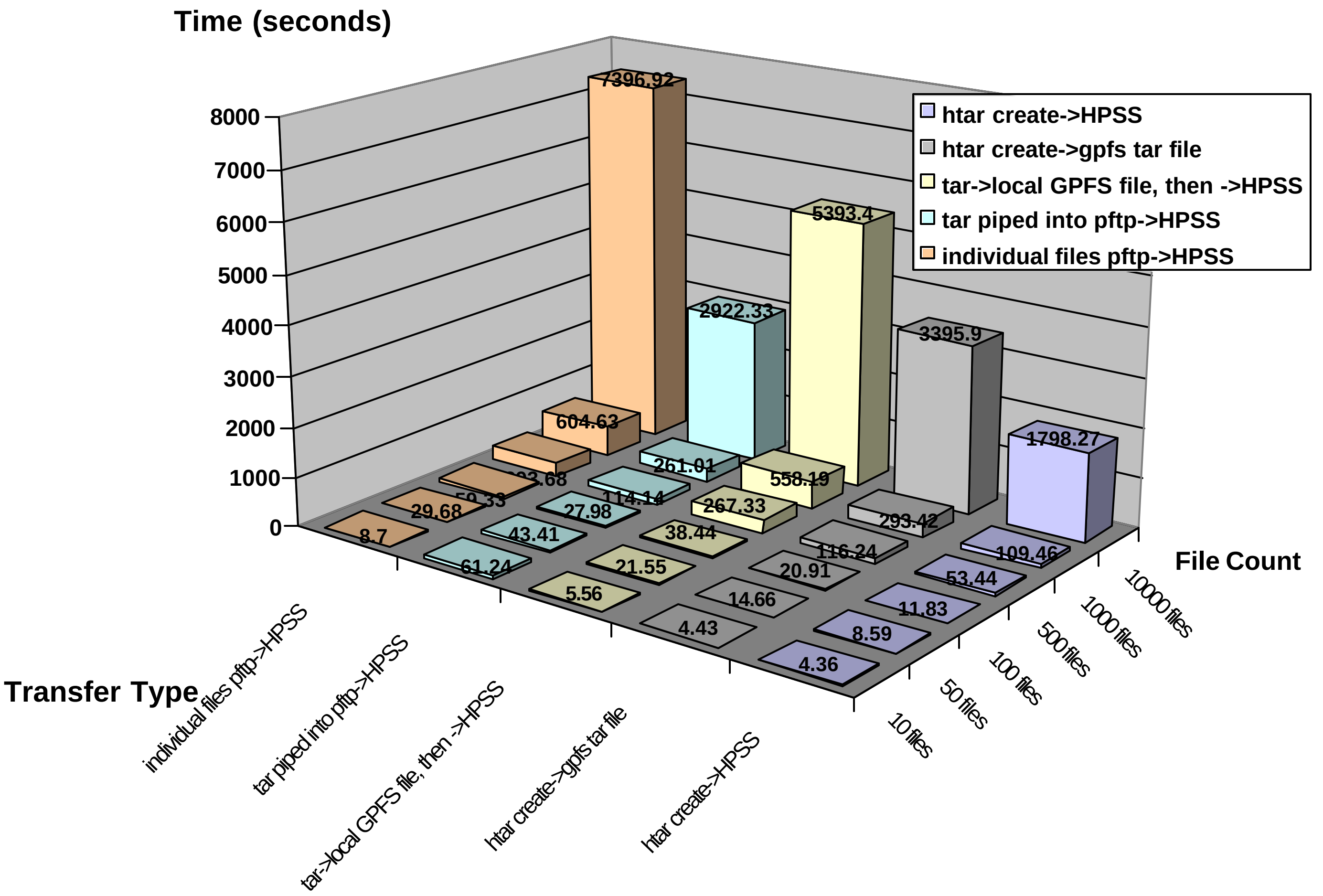
htar buffering overview (create)



htar performance testing results



More performance testing results



Future work

- Modification to work with ident based authentication
- File update function
- Append to existing archive function
- File delete function
- Archive file repack function